Fine Tuning:

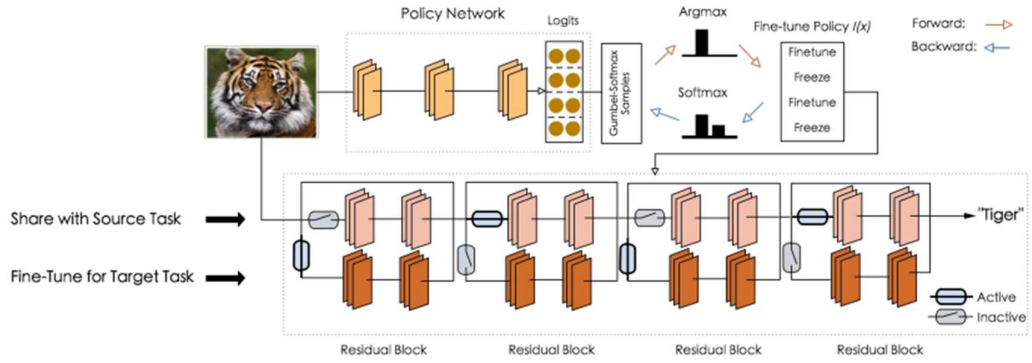
A process used on an already built learning / neural network model to perform a similar task that the model initially does.

ie: Updating *source task* model for a *target task*.
eg: Recognise a truck from a pool of vehicles by fine tuning a model already built to recognise cars.

i. Fine tuning pre-trained models have achieved impressive performance on standard NLP benchmarks. However, the resultant model generalizability remains poorly understood.
ii. In a standard CNN, we may want to keep weights of a few learning layers constant (frozen layers), and delete or add more layers (fine-tuned layers) as the task at hand demands.
iii. For example: Truck classification from a model that recognises cars may keep layers identifying door handles, headlights, windows etc the same, and add up layer(s) to identify a carriage, change weight to update tyre sizes etc.

Transfer Learning:

i. Pre-training a neural network model on a known task, for instance ImageNet, and then performing fine tuning – using the trained neural network as the basis of a new purpose-specific model.
ii. A learning methodology used to enhance learning by directly using simpler models already developed to perform similar tasks.
Fine tuning is a transfer learning methodology.
iii.



iv. SpotTune_Transfer_Learning_Through_Adaptive_Fine-Tuning_CVPR_2019_paper

Language Models:

1. BERT Model (Bi-directional Encoders Representation from Transformers):
   As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it's non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word).
   BERT in detail
2. N-gram Model:

i.

Intuitive Formulation

Let's start with equation P(w|h), the probability of word *w*, given some history, *h*. For example,

$$P(the \mid its\ water\ is\ so\ transperant\ that)$$

> *Here,*
> *w = The*
> *h = its water is so transparent that*

And, one way to estimate the above probability function is through the relative frequency count approach, where you would take a substantially large corpus, count the number of times you see ***its water is so transparent that***, and then count the number of times it is followed by ***the***. In other words, you are answering the question:

> *Out of the times you saw the history h, how many times did the word w follow it*

ii. It can be basically stated as the sequence of N words grouped together. Instead of counting probabilities of occurrence of every word in the corpus that leads to a word prediction, N-gram uses only the probabilities of a few historical words.

iii.

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$$

iv. The assumption that the probability of a word depends only on the previous word is also known as the ***Markov*** assumption. (under Bigram model)

v. Word prediction using Maximum Likelihood Estimation (MLE).

vi. Intro to N-Gram LM